```
Imports System.Drawing.Drawing2D
Public Class BarGraph
    Inherits System.Windows.Forms.UserControl
#Region "Variables and Enumerations"




    Private privatetotalLEDs As Integer = 20

    Private privatePanelHeight As Integer = CInt(Me.Height - 4)
    Private privatePanelWidth As Integer = CInt(Me.Width - 4)

    Private privateLEDHeight As Integer = CInt((Me.Height - 6) / (privatetotalLEDs + 2))
    Private privateLEDWidth As Integer = CInt(Me.Width - 6)

    Private privateLEDColor As System.Drawing.Color = Color.Gold

    Private privatePreviousLEDColor As System.Drawing.Color = Color.Gold
    Private privatePanelColor As System.Drawing.Color = Color.Black
    Private privatePanelBackColor As System.Drawing.Color = Color.Black

    Private privateMinimumValue As Integer = 0
    Private privateMaximumValue As Integer = 100

    Private privateBarGraphValue As Integer = 0
    Private privateBarGraphPreviousValue As Integer = 0

    Private privateLEDLeft As Integer = 5
    Private privateBarGraphStyle As Boolean = False ' True then incremental, False then absolute
    Private privateZeroShift As Integer = 0

#End Region

#Region "Control Properties"
    Public Property TotalLEDs() As Integer
        Get
            Return privatetotalLEDs
        End Get
        Set(ByVal value As Integer)
            privatetotalLEDs = value





            Me.Invalidate()
        End Set
    End Property

    ' The color property of the beads.
    Public Property LEDColor() As System.Drawing.Color
    Get
        Return privateLEDColor
    End Get
    Set(ByVal value As System.Drawing.Color)
        privateLEDColor = value
        'UpdateBarGraphColor()
        If Not (privatePreviousLEDColor = privateLEDColor) Then
            Me.Invalidate()
            privatePreviousLEDColor = privateLEDColor
        End If
    End Set
End Property

' The color property of the beads.
Public Property LEDLeft() As Integer
    Get
        Return privateLEDLeft
    End Get
    Set(ByVal value As Integer)
        privateLEDLeft = value
        'UpdateBarGraphColor()
        Me.Invalidate()
    End Set
End Property

'Public Property PanelColor() As System.Drawing.Color
'    Get
'        Return privatePanelColor
'    End Get
'    Set(ByVal value As System.Drawing.Color)
'        privatePanelColor = value
'        Panel1.BackColor = privatePanelColor
'        Me.Invalidate()
'    End Set
'End Property

Public Property PanelBackColor() As System.Drawing.Color
    Get
        Return privatePanelBackColor
    End Get
    Set(ByVal value As System.Drawing.Color)
        privatePanelBackColor = value
        Me.BackColor = privatePanelBackColor
        Me.Invalidate()
    End Set
End Property

' The number of beads on the control.
Public Property MinimumValue() As Integer
    Get
        Return privateMinimumValue
    End Get
```

```
        Set(ByVal value As Integer)
            If value < privateMaximumValue Then
                privateMinimumValue = value
            End If
            Me.Invalidate()
        End Set
    End Property

    ' The score displayed by the control.
    Public Property MaximumValue() As Integer
        Get
            Return privateMaximumValue
        End Get
        Set(ByVal value As Integer)
            If value > privateMinimumValue Then
                privateMaximumValue = value
            End If
            Me.Invalidate()
        End Set
    End Property
    Public Property BarGraphValue() As Integer
        Get
            Return privateBarGraphValue
        End Get
        Set(ByVal value As Integer)
            If (value <= privateMaximumValue) And (value >= privateMinimumValue) Then
                privateBarGraphValue = value



            End If
            If Not (privateBarGraphPreviousValue = privateBarGraphValue) Then
                UpdateBarGraphValue()
                Me.Invalidate()
            End If

        End Set
    End Property

    Public Property BarGraphStyle() As Boolean
        Get
            Return privateBarGraphStyle
        End Get
        Set(ByVal value As Boolean)
            ' True then incremental, False then absolute
            privateBarGraphStyle = value
            UpdateBarGraphValue()

            Me.Invalidate()
        End Set
    End Property
```

```
#End Region

#Region "Drawing Functions"
    Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)

        Dim rect As System.Drawing.Rectangle = e.ClipRectangle
        Dim g As Graphics = e.Graphics
        Dim mainPen As New Pen(Color.Black)


        UpdateBarGraphValue()
    End Sub



#End Region

#Region "Event Handlers"



#End Region

    Protected Overrides Sub Finalize()
        MyBase.Finalize()
    End Sub

    Private Sub UpdateBarGraphValue()
        Dim i As Integer
        Dim foo_int As Single
        Dim foo_int1 As Integer
        Dim foo_int2 As Single
        'Dim foo_boolean(20) As Boolean
        foo_int = privateMaximumValue - privateMinimumValue 'Find total range of bargraph in this
case 100-0=100
        foo_int = foo_int / privatetotalLEDs 'find value associated with each LED in this case
100/20=5
        foo_int2 = (privateBarGraphValue / foo_int) 'find total LEDs to be visible in this case 50/5=10
        If foo_int2 < 0 Then
            foo_int1 = CInt(foo_int2 * (-1.0))
        Else
            foo_int1 = CInt(foo_int2)
        End If

        'Set Panel Width
        privatePanelWidth = CInt(Me.Width)

        'Set Panel Height
        privatePanelHeight = CInt(Me.Height)
        'Set LED Width
        privateLEDWidth = CInt(((privatePanelWidth / 2) - (privatePanelWidth / 10)) * 2)

        'Set left position of the LEDs
        privateLEDLeft = (privatePanelWidth - privateLEDWidth) / 2

        'Set LED Height
        privateLEDHeight = CInt((privatePanelHeight - (6 + privatetotalLEDs)) / (privatetotalLEDs))
        Dim s1 As Single = CSng(privateLEDHeight)
        ' Create pen.
        'Dim myPen As New System.Drawing.Pen(privateLEDColor, s1)
        Dim formGraphics As System.Drawing.Graphics
        Dim brush As System.Drawing.Brush = New SolidBrush(privateLEDColor)

        formGraphics = Me.CreateGraphics()
        Dim Yi As Integer
        If privateBarGraphStyle = True Then ' if incremental
            If (privateBarGraphValue > 0) Then
                Yi = ((privatePanelHeight - 3) - (privateLEDHeight * (CInt((privatetotalLEDs / 2) - 1) +
1)) - (CInt((privatetotalLEDs / 2) - 1) + 3))
                For i = 0 To (foo_int1 - 1)
                    Yi = Yi - (privateLEDHeight + 1)

                    formGraphics.FillRectangle(brush, privateLEDLeft, Yi, privateLEDWidth,
privateLEDHeight)

                    'formGraphics.DrawLine(myPen, privateLEDLeft, Yi, privateLEDWidth, Yi)
                Next
            Else
                Yi = ((privatePanelHeight - 3) - (privateLEDHeight * (CInt((privatetotalLEDs / 2) - 1) +
1)) - (CInt((privatetotalLEDs / 2) - 1) + 3))
                For i = 0 To (foo_int1 - 1)
                    Yi = Yi + (privateLEDHeight + 1)
                    formGraphics.FillRectangle(brush, privateLEDLeft, Yi, privateLEDWidth,
privateLEDHeight)
                    'formGraphics.DrawLine(myPen, privateLEDLeft, Yi, privateLEDWidth, Yi)
                Next
            End If

        Else
            For i = 0 To (foo_int1 - 1)
                Yi = ((privatePanelHeight - 3) - (privateLEDHeight * (i + 1)) - (i + 3))
                formGraphics.FillRectangle(brush, privateLEDLeft, Yi, privateLEDWidth,
privateLEDHeight)
                'formGraphics.DrawLine(myPen, privateLEDLeft, Yi, privateLEDWidth, Yi)
            Next
        End If
        brush.Dispose()
        'myPen.Dispose()
        formGraphics.Dispose()
        privateBarGraphPreviousValue = privateBarGraphValue
    End Sub
End Class
```